

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method comprising:
storing native code associated with a first method within a native code space;
creating a symbolic reference to the first method in a method table;
determining whether the native code space exceeds a threshold in response to an
invocation of a second method;
unwinding a stack to determine which methods are active;
reclaiming the native code associated with the first method and compiling byte
code into native code associated with the second method in response to
determining that the second method is active~~the determination~~; and
updating the method table for the first method to reference an appropriate
symbolic reference.
2. (Previously Presented) The method as set forth in claim 2, wherein reclaiming the
native code associated with the first method and compiling byte code into native
code associated with the second method in response to the determination
comprises reclaiming the native code associated with the first method in response
to a determination that the native code space exceeds the threshold.
3. (Previously Presented) The method as set forth in claim 2, further comprising
storing the native code associated with the second method within the native code
space in response to the compilation.

4. (Previously Presented) The method as set forth in claim 2, further comprising:
invoking the first method following the reclamation; and
re-compiling byte code into the native code associated with the first method in
response to the invocation of the first method.
5. (Previously Presented) The method as set forth in claim 2, wherein reclaiming the
native code associated with the first method and compiling byte code into native
code associated with the second method in response to the determination
comprises compiling byte code into native code associated with the second
method.
6. (Previously Presented) The method as set forth in claim 5, wherein compiling
byte code into native code associated with the second method comprises
compiling byte code into native code associated with the second method utilizing
a just-in-time compiler.
7. (Previously Presented) The method as set forth in claim 2, wherein reclaiming the
native code associated with the first method and compiling byte code into native
code associated with the second method in response to the determination
comprises:
determining whether the first method is active or inactive; and
reclaiming the native code associated with the first method in response to a
determination that the first method is inactive.

8. (Previously Presented) The method as set forth in claim 7, further comprising:
reclaiming the native code associated with the first method and compiling byte
code into native code associated with the second method in response to the
determination further comprises determining whether the first method is
hot or cold in response to a determination that the first method is inactive;
and
reclaiming the native code associated with the first method in response to a
determination that the first method is inactive comprises reclaiming the
native code associated with the first method in response to a determination
that the first method is cold.
9. (Currently Amended) A ~~data processing system readable~~ machine-readable
medium having stored thereon data representing sets of instructions which, when
executed by a machine, cause the machine to a plurality of instructions executable
~~by a data processing system embodied therein, wherein the plurality of~~
~~instructions when executed cause the data processing system to perform~~
~~operations comprising:~~
~~storing~~ store native code associated with a first method within a native code
space;
~~creating~~ create a symbolic reference to the first method in a method table;
~~determining~~ determine whether the native code space exceeds a threshold in
response to an invocation of a second method;
unwind a stack to determine which methods are active;

~~reclaiming~~ reclaim the native code associated with the first method and compiling byte code into native code associated with the second method in response to determining that the second method is active ~~the determination~~; and ~~updating~~ update the method table for the first method to reference an appropriate symbolic reference.

10. (Currently Amended) The ~~data processing system readable~~ machine-readable medium of claim 9, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination comprises reclaiming the native code associated with the first method in response to a determination that the native code space exceeds the threshold.
11. (Currently Amended) The ~~data processing system readable~~ machine-readable medium of claim 9, wherein the sets of instructions, when executed by the machine, further cause the machine ~~the plurality of instructions when executed further cause the data processing system~~ to perform operations comprising storing the native code associated with the second method within the native code space in response to the compilation.
12. (Currently Amended) The ~~data processing system readable~~ machine-readable medium of claim 9, the sets of instructions, when executed by the machine, further cause the machine ~~wherein the plurality of instructions when executed further cause the data processing system~~ to perform operations comprising invoking the first method following the reclamation; and re-compiling byte code

into the native code associated with the first method in response to the invocation of the first method.

13. (Currently Amended) The ~~data-processing-system-readable~~machine-readable medium of claim 9, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination comprises compiling byte code into native code associated with the second method.
14. (Currently Amended) The ~~data-processing-system-readable~~machine-readable medium of claim 13, wherein compiling byte code into native code associated with the second method comprises compiling byte code into native code associated with the second method utilizing a just-in-time compiler.
15. (Currently Amended) The ~~data-processing-system-readable~~machine-readable medium of claim 9, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination comprises:

determining whether the first method is active or inactive; and

reclaiming the native code associated with the first method in response to a

determination that the first method is inactive.
16. (Currently Amended) The ~~data-processing-system-readable~~machine-readable medium of claim 15, wherein the sets of instructions, when executed by the machine, further cause the machine to further comprising:

~~reclaiming~~ reclaim the native code associated with the first method and
compiling byte code into native code associated with the second method
in response to the determination further comprises determining whether
the first method is hot or cold; and
~~reclaiming~~ reclaim the native code associated with the first method in response to
a determination that the first method is inactive comprises reclaiming the
native code associated with the first method in response to a determination
that the first method is cold.

17. (Previously Presented) A data processing system comprising:
- a storage device;
 - a processor coupled with the storage device, the processor to process data and
execute instructions; and
 - a memory coupled with the storage device and the processor, the memory to store
data including a plurality of instructions which when executed by the
processor cause the data processing system to perform operations having:
storing native code associated with a first method within a native code
space of the memory;
creating a symbolic reference to the first method in a method table;
determining whether the native code space exceeds a threshold in response
to an invocation of a second method;
unwinding a stack to determine which methods are active;
reclaiming the native code associated with the first method and compiling
byte code into native code associated with the second method in
response to determining that the second method is active ~~the~~
~~determination~~; and

updating the method table for the first method to reference an appropriate symbolic reference.

18. (Previously Presented) The data processing system of claim 17, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination comprises reclaiming the native code associated with the first method in response to a determination that the native code space exceeds the threshold.
19. (Previously Presented) The data processing system of claim 17, wherein the plurality of instructions when executed further cause the data processing system to perform operations comprising storing the native code associated with the second method within the native code space in response to the compilation.
20. (Previously Presented) The data processing system of claim 17, wherein the plurality of instructions when executed further cause the data processing system to perform operations comprising invoking the first method following the reclamation; and re-compiling byte code into the native code associated with the first method in response to the invocation of the first method.
21. (Previously Presented) The data processing system of claim 17, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the

determination comprises compiling byte code into native code associated with the second method.

22. (Previously Presented) The data processing system of claim 21, wherein compiling byte code into native code associated with the second method comprises compiling byte code into native code associated with the second method utilizing a just-in-time compiler.
23. (Previously Presented) The data processing system of claim 17, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination comprises:

determining whether the first method is active or inactive; and

reclaiming the native code associated with the first method in response to a determination that the first method is inactive.
24. (Previously Presented) The data processing system of claim 23, further comprising:

reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination further comprises determining whether the first method is hot or cold; and

reclaiming the native code associated with the first method in response to a determination that the first method is inactive comprises reclaiming the

native code associated with the first method in response to a determination that the first method is cold.

Claims 25-28 (Cancelled)